What is claimed is:

1. A method for synchronizing a service clock at a destination node with a service clock at a source node for circuit emulation service over a packet network, the method comprising:

receiving data packets from a source node at at least one port of the destination node;

at the destination node, removing from the data packets residual time stamp (RTS) values that were created at the source node based on at least the service clock at the source node;

determining a majority count and a minority count, based on the RTS values, for each of a plurality of time periods; and

using the majority and minority counts for the plurality of time periods to calculate a control value to set the frequency of a service clock at the destination node for use in receiving data packets.

2. The method of claim 1, wherein determining a majority count and a minority count comprises counting values based on the two's complement of a difference between consecutive RTS values.

3. The method of claim 1, wherein determining a majority count and a minority count includes counting the occurrence of three expected values.

4. The method of claim 1, wherein determining a majority count and a minority count comprises determining a majority count and a minority count over four time periods.

5. The method of claim 4, wherein determining a majority count and a minority count over four time periods comprises determining a majority count and a minority count over 2, 20, 200 and 2000 second time periods.

6. The method of claim 1, wherein determining a majority count and a minority count over a plurality of time periods includes reading a plurality of counters every 2 seconds.

7. The method of claim 1, wherein using the majority and minority counts for the plurality of time periods to calculate a control value comprises:

comparing calculations for the control value based on the majority and minority counts for the plurality of time periods; and

selecting one of the calculated control values generated from the majority and minority counts from one of the time periods.

8. The method of claim 7, wherein comparing calculations for the control value based on the majority and minority counts for the plurality of time periods comprises sequentially comparing calculations for the control value for a longest of the plurality of time periods with the control values for the other of the plurality of time periods.

9. The method of claim 8, wherein selecting one of the calculated control values comprises selecting the control value for the longest of the plurality of time periods unless one of the control values of the other of the plurality of time periods is not within selected bounds of the control value of the longest of the plurality of time periods.

10. The method of claim 1, and further monitoring the fill level of a buffer associated with the at least one port of the destination node to determine whether to adjust the frequency of the service clock.

11. The method of claim 10, wherein:

when the average fill level decreases over time, decreasing the frequency of the service clock at the destination node; and

when the average fill level increases over time, increasing the frequency of the service clock at the destination node.

12. The method of claim 10, wherein monitoring the fill level of the buffer comprises averaging the buffer fill level of each of the ports of the destination node for at least 10 seconds.

13. The method of claim 1, using the majority and minority counts for the plurality of time periods to calculate a control value to set the frequency of a service clock at the destination node for use in receiving data packets comprises calculating an n-bit number, X, for a direct digital synthesis circuit using the following equation:

$$X = \frac{2^n}{f_{REF}} F(R'TS)$$

wherein *F(RTS')* is a function that relates the frequency at the source node to the stored RTS values over a period of time.

14. A method for synchronizing a service clock at a destination node in a packet switch network wherein packets are transmitted from a source node to a destination node and wherein the source node calculates and transmits residual time stamp (RTS) values in the data packets, the method comprising:

removing RTS values from data packets at the destination node; and

using RTS values over a plurality of time periods to set a direct digital synthesis circuit to act as the service clock for the destination node with a frequency that is substantially synchronized with a service clock at the source node.

15. The method of claim 14, wherein using RTS values over a plurality of time periods to set a direct digital synthesis circuit comprises determining a majority count and a minority count, based on the RTS values, over each of a plurality of time periods and using the majority and minority counts for the plurality of time periods to calculate a control value for the direct digital synthesis circuit.

16. The method of claim 15, wherein using the majority and minority counts for the plurality of time periods to calculate a control value comprises:

comparing calculations for the control value based on the majority and minority counts for the plurality of time periods; and

selecting one of the calculated control values generated from the majority and minority counts from one of the time periods.

17. The method of claim 16, wherein comparing calculations for the control value based on the majority and minority counts for the plurality of time periods comprises comparing calculations for the control value for a longest of the plurality of time periods with the control values for the other of the plurality of time periods.

18. The method of claim 17, wherein selecting one of the calculated control values comprises selecting the control value for the longest of the plurality of time periods unless one of the control values of the other of the plurality of time periods is not within selected bounds of the control value of the longest of the plurality of time periods.

19. The method of claim 14, wherein using the RTS values to set the direct digital synthesis circuit comprises generating an n-bit number and providing the n-bit number to the direct digital synthesis circuit.

20. The method of claim 19, wherein the destination node calculates the value of the n-bit number using non-floating point calculations.

21. The method of claim 14, and further monitoring the fill level of a buffer associated with the at least one port of the destination node to determine whether to adjust the frequency of the service clock.

22. The method of claim 21, wherein:

when the average fill level decreases over time, decreasing the frequency of the service clock at the destination node; and

when the average fill level increases over time, increasing the frequency of the service clock at the destination node.

23. The method of claim 21, wherein monitoring the fill level of the buffer comprises averaging the buffer fill level of each of the ports of the destination node for at least 10 seconds.

24. A system for recovering a service clock at a network node for circuit emulation service over a packet network, the system comprising:

a direct digital synthesis circuit for each port that generates a local service clock signal for each port of the network node;

a circuit, coupled to a port of the network node, that removes residual time stamp (RTS) values from data packets that are received at the port of the network node;

a counting circuit that determines majority and minority counts based on the RTS values over a plurality of time periods; and

a microcontroller that uses the majority and minority counts over the plurality of time periods to generate a number to set the frequency of the direct digital synthesis circuit.

25. The system of claim 24, and further comprising a buffer associated with each port, wherein the microcontroller further monitors the fill level of the buffer to determine whether to adjust the frequency of the direct digital synthesis circuit for each port.

26. The system of claim 24, wherein the microcontroller compares calculations for the number for the direct digital synthesis circuit based on the majority and minority counts for the plurality of time periods and selects one of the calculated numbers generated from the majority and minority counts from one of the time periods.

27. The system of claim 24, wherein the microcontroller reads values from the counting circuit every two seconds.

28. The system of claim 24, wherein the microcontroller tracks majority and minority counts for four time periods.

29. The system of claim 28, wherein the four time periods comprise 2, 20, 200, and 2000 seconds.

30. The system of claim 24, wherein the counting circuit includes counters for each of three expected RTS values.

31. The system of claim 24, wherein the circuit that removes the residual time stamp values provides the two's complement of the difference between successive RTS values to the counting circuit.

32. A method for service clock recovery, the method comprising:

removing RTS values from data packets at the destination node; and

using the removed RTS values over a plurality of time periods to set the service clock for the destination node.

33. The method of claim 32, wherein using the removed RTS values comprises using majority and minority counts based on RTS values over at least two time periods.

34. The method of claim 32, wherein using the removed RTS values comprises calculating a plurality of control signal based on RTS values received over a plurality of time periods and selecting one of the control signals to control the local service clock.

35. The method of claim 32, and further using buffer fill levels to control the local service clock.

36. A network node coupleable to a packet network, the network node comprising:

a packet disassembler having an input coupleable to the packet network;

an overhead processor, coupled to the packet disassembler, that removes data and residual time stamp (RTS) values from data packets received at the input;

a counting circuit, coupled to the overhead processor, that determines majority and minority counts based on the RTS values over a plurality of time periods;

a data buffer, coupled to the overhead processor, that receives the data from the packets;

a line interface unit, coupled to the data buffer;

a direct digital synthesis circuit, coupled to the data buffer and the line interface unit, that generates a local service clock signal; and

a microcontroller, responsive to the counting circuit, wherein the microcontroller uses the majority and minority counts from the counting circuit over the plurality of time periods to generate a number to set the frequency of the direct digital synthesis circuit.

37. The network node of claim 36, wherein the microcontroller is further responsive to a fill level of the data buffer to adjust the control signal provided to the direct digital synthesis circuit.

38. The network node of claim 36, wherein the microcontroller uses the majority and minority counts over time periods of 2, 20, 200, and 2000 seconds to generate the control signal for the direct digital synthesis circuit.

39. The network node of claim 36, wherein the microcontroller reads majority and minority counts from the counting circuit every 2 seconds.

40. The network node of claim 36, wherein the microcontroller calculates control signals based on majority and minority counts for each of the plurality of time periods and selects one of the control signals to provide to the direct digital synthesis circuit.

41. A method for controlling a local clock in a node coupled to a packet network, the method comprising:

generating a first control signal based on RTS values received during a first period of time;

generating at least one additional control signal based on RTS values received during at least one additional period of time; and

selecting one of the first and at least one additional control signals to control the frequency of a local clock.

42. A network node coupleable to a packet network, the network node comprising:

a packet disassembler having an input coupleable to the packet network;

an overhead processor, coupled to the packet disassembler, that removes data and residual time stamp (RTS) values from data packets received at the input;

a counting circuit, coupled to the overhead processor, that produces at least two counts based on the RTS values for a plurality of time periods;

a data buffer, coupled to the overhead processor, that receives the data from the packets;

a line interface unit, coupled to the data buffer;

a direct digital synthesis circuit, coupled to the data buffer and the line interface unit, that generates a local service clock signal; and

a microcontroller, responsive to the counting circuit, wherein the microcontroller uses the at least two counts from the counting circuit for the plurality of time periods to generate at least two control values to set the frequency of the direct digital synthesis circuit, selects one of the at least two control values, and provides the selected control

value to the direct digital synthesis circuit.

From 100.118:

43. A method for clock recovery in a packet network, the method comprising:

receiving data packets at a destination node;

storing data from the data packets in a buffer;

reading the data packets out of the buffer using a locally generated clock;

monitoring a fill level of the buffer over a plurality of time periods;

identifying a relative maximum fill level for the buffer during each time period; and

selectively using the relative maximum fill levels for the plurality of time periods to control a frequency of the locally generated clock so as to control the rate at which data is read out of the buffer.

44. The method of claim 43, wherein selectively using the relative maximum fill level comprises selecting a control signal based on the maximum buffer fill level of one of the plurality of time periods.

45. The method of claim 43, wherein selectively using the relative maximum fill level comprises:

comparing calculations for the control value based on the relative maximum buffer fill level for the plurality of time periods; and

selecting one of the calculated control values generated from the relative maximum buffer fill level from one of the time periods.

46. The method of claim 45, wherein comparing calculations for the control value based on the relative maximum buffer fill level for the plurality of time periods comprises comparing calculations for the control value for a longest of the plurality of time periods with the control values for the other of the plurality of time periods.

47. The method of claim 45, wherein selecting one of the calculated control values comprises selecting the control value for the longest of the plurality of time periods unless one of the control values of the other of the plurality of time periods is not within selected bounds of the control value of the longest of the plurality of time periods.

48. The method of claim 43, wherein monitoring a fill level of the buffer comprises monitoring a difference between a read address and a write address of the buffer.

49. A node in a telecommunications network, comprising;

a buffer having an input that is adapted to receive data packets from another node;

a peak fill level detector, responsive to address signals for the buffer, wherein the peak fill level detector includes a register that stores relative peak fill levels acquired during a plurality of time periods;

a variable oscillator coupled to the buffer that controls the rate at which data is processed in the node; and

a processor coupled to the peak fill level detector, wherein the processor receives the relative peak fill levels from the register and uses the relative peak fill levels for the plurality of time periods to generate a control value that selectively adjusts the variable oscillator.

50. The node of claim 49, wherein the peak fill level detector includes a register that is continuously updated over a period of time with a relative maximum buffer fill level.

51. The node of claim 49, wherein the peak fill detector comprises a mechanism that compares a read address and a write address for the buffer and stores the maximum buffer fill level observed over a period of time.

52. The node of claim 49, wherein the variable oscillator comprises a numerically controlled oscillator.

53. The node of claim 49, wherein the processor is programmed to implement a method for selectively using the relative peak fill levels, the method comprising:

comparing calculations for the control value based on the relative peak fill level for the plurality of time periods; and

selecting one of the calculated control values generated from the relative peak fill level from one of the time periods.

54. The node of claim 53, wherein comparing calculations for the control value based on the relative peak fill level for the plurality of time periods comprises comparing calculations for the control value for a longest of the plurality of time periods with the control values for the other of the plurality of time periods.

55. The node of claim 54, wherein selecting one of the calculated control values comprises selecting the control value for the longest of the plurality of time periods unless one of the control values of the other of the plurality of time periods is not within selected bounds of the control value of the longest of the plurality of time periods.

56. A method for adaptive clock recovery, the method comprising:

monitoring a buffer fill level for a plurality of time periods;

identifying a relative maximum fill level during the plurality of time periods; and

controlling the frequency of a recovered clock signal based on the relative maximum fill levels for the plurality of time periods such that the recovered clock signal is substantially free of jitter.

57. The method of claim 56, wherein controlling the frequency of the recovered clock signal comprises:

determining a rate of change in the relative maximum fill level for each time period; and

calculating a numerical value for a numerically controlled oscillator to compensate for the rate of change for a selected time period.

58. The method of claim 56, wherein controlling the frequency of the recovered clock signal comprises:

comparing calculations for a control value based on the relative maximum fill level for the plurality of time periods; and

selecting one of the calculated control values generated from the relative maximum fill level from one of the time periods.

59. The method of claim 58, wherein comparing calculations for the control value based on the relative maximum fill level for the plurality of time periods comprises comparing calculations for the control value for a longest of the plurality of time periods with the control values for the other of the plurality of time periods.

60. The method of claim 59, wherein selecting one of the calculated control values comprises selecting the control value for the longest of the plurality of time periods unless one of the control values of the other of the plurality of time periods is not within selected bounds of the control value of the longest of the plurality of time periods.

61. A method for recovering a clock at a destination node, the method comprising:

receiving data packets at the destination node;

calculating control values for a numerically controlled oscillator over a plurality of time periods; and

selectively using the control values to set the frequency of the numerically controlled oscillator.

62. The method of claim 61, wherein calculating control values comprises calculating control values using a peak buffer fill level for each of the plurality of time periods.

63. The method of claim 61, wherein calculating control values comprises calculating control values using residual time stamp (RTS) values.